GitHub Username: edcorners

GameSight

Description

Discover and keep track of your favorite games coming out soon. See full game information including videos, reviews, platforms and publisher. Keep track of progress at games you own. Receive notifications when games are launched.

Intended User

Gamers interested on keeping track of new game launches and game progress.

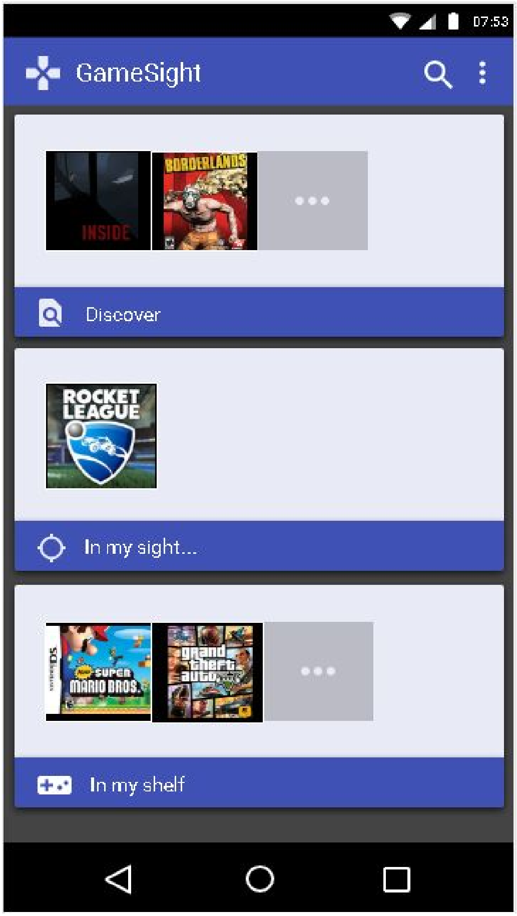
Features

- Track new releases of video games
- Filter search results or local collections by console
- Receive notifications when release dates of your tracked games are close
- Track progress of your collection of games
- See your game collection offline

User Interface Mocks

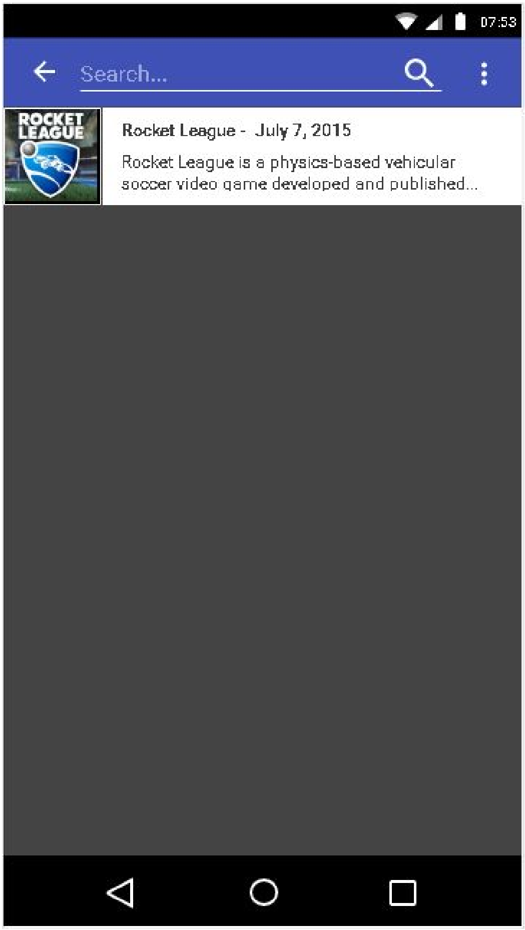
Screen 1- Main

On the main screen users will be able to see a preview of the three main collections in the app; new game suggestions (**Discover)**, games the user is tracking (**In my sights…**) and games the user is playing (**In my shelf**):



Screen 2 - Search

When the user taps the “**Search**” text/icon on the Main screen (or secondary screens), a search screen is displayed.



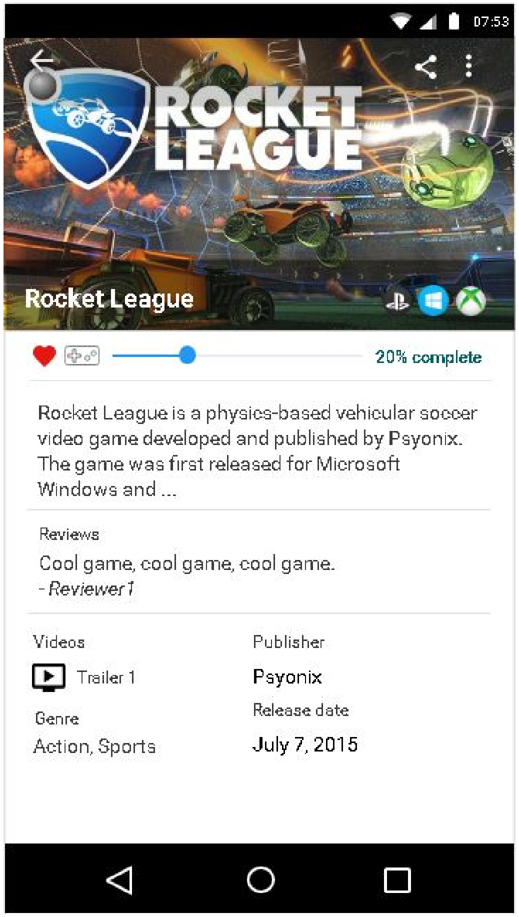
Rocket League - July 7, 2015

Rocket League is a physics-based vehicular soccer video game developed and published...

If the user taps on a result item the “**Details**” screen comes up. The user will also be able to go back to the previous screen.

Screen 3 - Details

The “**Details**” screen will present the user with detailed information of the game; description, release date, videos, genre, publisher and platforms. The user will be able to add the current game to the local collection using the heart icon. If the game is not released yet it will appear in the “**In my sight**” collection on the main screen. Otherwise it will appear in the “**In my shelf**” collection. If the game is in the “**In my shelf**” collection the user will be able to update game progress using a slider bar and share it to social networks.



Rocket League

20% complete

Rocket League is a physics-based vehicular soccer video game developed and published by Psyonix. The game was first released for Microsoft Windows and ...

Reviews

Cool game, cool game, cool game.
- *Reviewer1*

Videos

Trailer 1

Genre

Action, Sports

Publisher

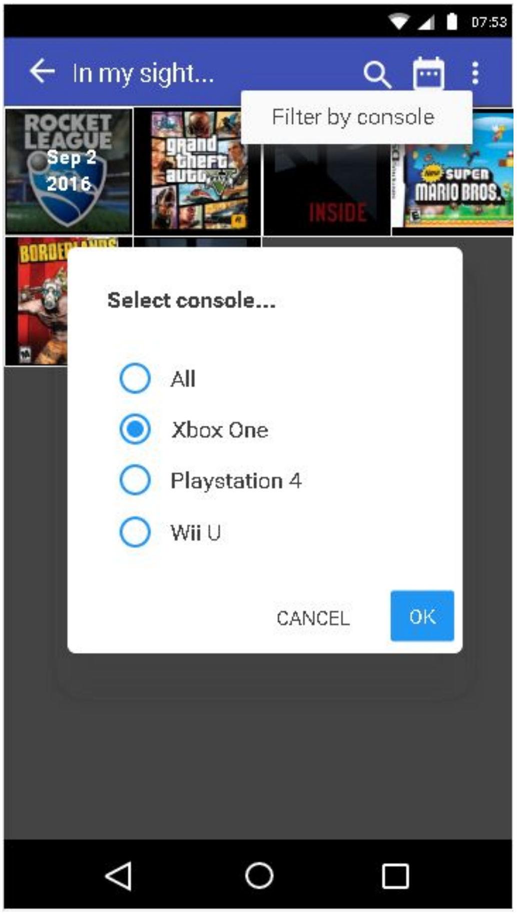
Psyonix

Release date

July 7, 2015

Screen 4a - Games grid (In my sight…)

The “**In my sight**” screen will appear when the user taps the icon/text on the main screen. In this screen the user will be able to see a grid with all the upcoming games in his local list. Games will be sorted by release date by default from nearest to farthest. Also, a calendar icon will allow the user to see the release date for the games overlaid on the cover picture. The user will be able to filter by console, go to the search screen and back to the main screen.



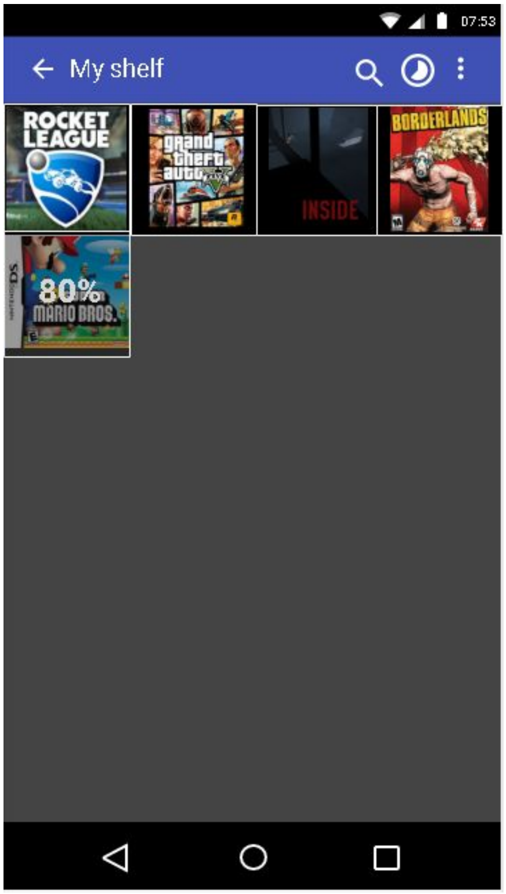
Select console...

○ All
● Xbox One
○ Playstation 4
○ Wii U

Tapping on the “**Discover**” card will present the user with the same layout as the “**In my sight**” screen, but with games the user might be interested in.

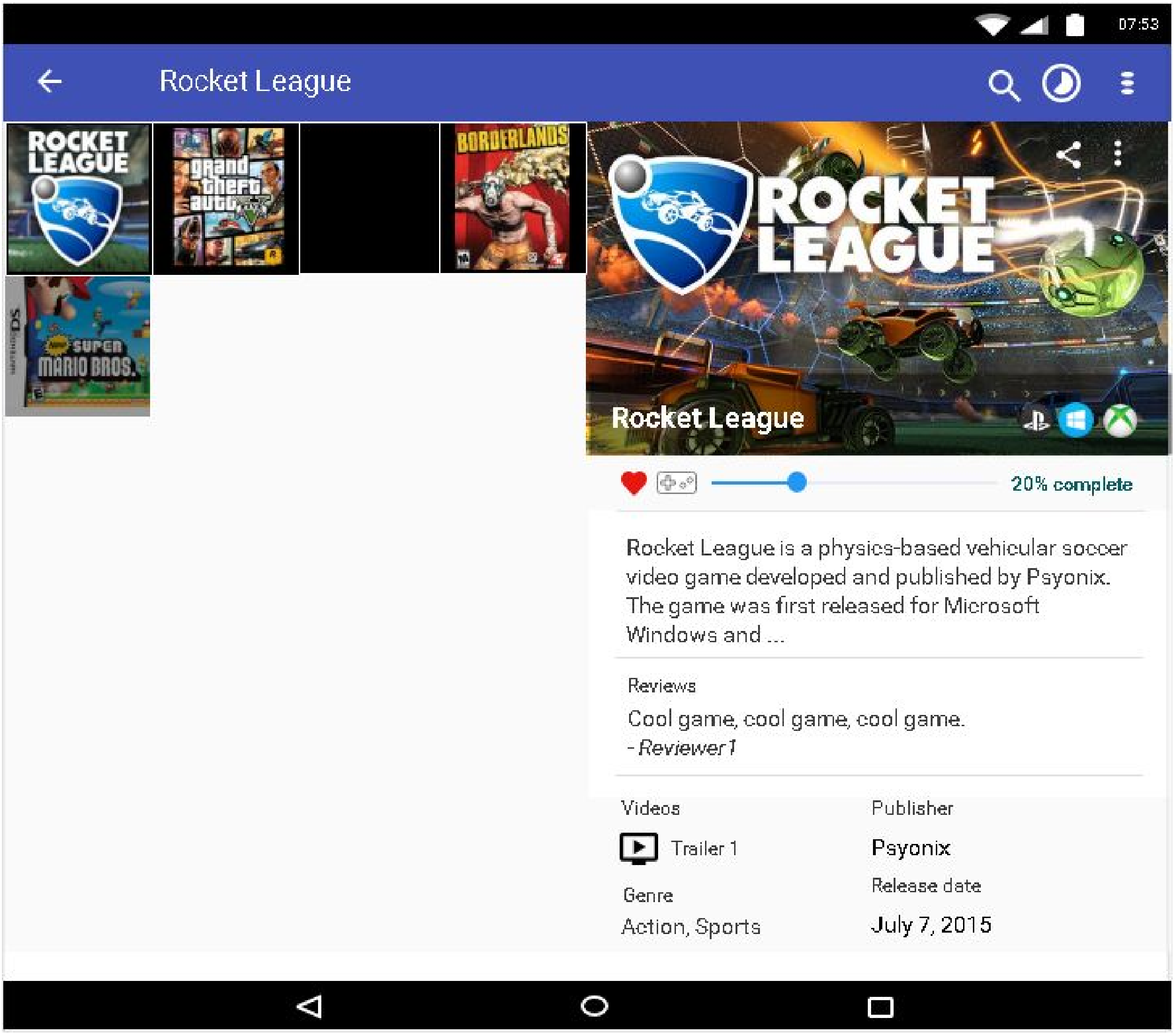
Screen 4b - Games grid (My shelf)

The “**My shelf**” screen will appear when the user taps the icon/text on the main screen. In this screen the user will be able to see a grid with games he is currently playing. Games will be sorted by release date by default from nearest to farthest. Also, a progress icon will allow the user to see the progress on each game overlaid on the cover picture. The user will be able to filter by console, go to the search screen and back to the main screen.



Screen 5 - Tablet view

Tablet view will display game grids and details side by side when in landscape mode.



Rocket League is a physics-based vehicular soccer video game developed and published by Psyonix. The game was first released for Microsoft Windows and ...

Reviews

Cool game, cool game, cool game.
- *Reviewer1*

Videos
Trailer 1

Genre
Action, Sports

Publisher
Psyonix

Release date
July 7, 2015

Key Considerations

Data persistence

- A content provider will provide data for locally stored games. That is, games that the user is currently playing or games the user is tracking for launch.
  The discover section will only display information if the user is connected to the internet.

Corner cases

- The user will be able to return to the main page from “Discover”, “In my sight” or “My shelf” using a back button on the top left corner of the screen.
- The “Details” view return the user to the previous screen after tapping the back button.
- Videos will be opened in an external browser or the youtube app according to what the API provides.
- Filter menu will open on a pop up and return to the parent screen after the user selects an option.

Libraries

- **Picasso** will be used for image loading since it provides a simple and efficient way to solve this problem.
- For the creation of a content provider the app will include **Schematic** to help quickly generate and maintain one.
- **Butterknife** will be used to ease binding UI elements to activity view objects.
- **Retrofit** will be used to quickly implement a client to consume the video games API .
- **GSON-converter** will be used as a converter from JSON to object for Retrofit.
- **Stetho** will be used to monitor the database and network requests.
- **Android support libraries** will be used as follows:
  - cardview: in order to implement cards in the main view ui
  - appcompat, support: in order to maintain retrocompatibility
  - design: in order to implement material design
  - recyclerview: in order to display the grid of games
  - support-annotations: in order to define enumerated lists
  - palette: to handle colors according to pictures

Google Play Services

- Google play analytics will be implemented to track most followed games for a future improvement on searches and a popular games section of the app.
- Free and paid versions will be implemented. Admob will be used to display ads on the free version.

Required Tasks

Task 1: Project Setup

- Initialize project in Android Studio and GitHub.
- Define dependencies (Libraries).
- Generate main activity, game grid (Discover, In my sight, In my shelf) activity, search and detail activities along with their fragments.

Task 2: Implement Content Provider and API Client

- Configure retrofit and GSON; create object model to support reading the games api through rest services.
- Implement content provider using Schematic. Define endpoints.

Task 3: Implement UI for Main Activity and Fragment

- Build UI for main screen. Create cards and recycler views for the collection of games. Define an item layout. Define navigation to detail, menu and search views.
- Build fragment/activity for main screen. Create Loader, recyclerview.

Task 4: Implement UI for Search view

- Build UI for the search view. Create result item view and search bar.
- Build activity for search view. Implement search routine and results display logic.

Task 5: Implement UI for Game Grid Activity and Fragment

- Build UI for Collection views. Create recyclerview for the grid of games. Build menu items for search, overlay information and filter by console.
- Build fragment/activity for the collection views screen. Create Loader, back recyclerview and menu options.

Task 6: Implement UI for Detail Activity and Fragment

- Build UI for Detail view. Create recyclerview for the grid of games. Build add to collection button, progress bar and game details.
- Build fragment/activity for the collection views screen. Create Loader, back recycler view and share, add to collection and progress bar buttons.

Task 7: Implement UI for tablets

- Build UI for tablets. Adjust activities/fragments and create tablet layouts.

Task 8: Implement widget

- Build a widget that displays games on “In my sight” collection by “nearest to launch date” order.

Task 9: Implement Analytics and Admob

- Implement paid and free versions. Free using ads. Implement analytics to track most popular games.

Task 10: Implement sharing and notifications

- Implement sharing game progress to social networks and notifications for game releases.

Task 11: Final Review

- Ensure RTL, internationalization, content description, notifying of errors and material design best practices.
- Test for bugs and fix.
- Build signed apk.